7. Web-Based Applications

The DII COE includes a collection of COE-component segments to support Web-based applications. This provides a foundation for the development of Web-based segments within the DII COE, and for mission applications built on top of the COE. The Web component segments provide services and infrastructure for the delivery of HTML files[1] from a Web server to a Web browser. One of the key goals in adding Web capabilities to the DII COE is to foster sufficient discipline to prevent anarchy, while permitting a flexible Web runtime environment.

The COE Web component segments are designed to meld diverse system and operator requirements, and leverage advances in Internet technology and functionality. Evolution of Web component segments is driven by several factors:

- architectural freedom for creativity and rapid progress,
- customer demand for access to (and sharing of) remote data sources, and
- the rapid pace of Web innovation.

This chapter is devoted to explaining the COE Web component segments, and to provide implementation guidance for creating Web mission-application segments. It should be noted that the majority of users will likely use PCs, so this is considered the target client platform for Web development. However, the principles and techniques presented here work equally well for the Unix environment.

Section 7.1 discusses fundamental COE Web concepts. Section 7.2 describes Web administration and user accounts. Section 7.3 contains miscellaneous information pertinent to developing Web segments, including an overview of HTML requirements for the COE Web. Section 7.4 describes what happens when Web segments are installed, and section 7.5 completes the chapter with a brief discussion of supported configurations.

[1] The term "HTML file" is used throughout this chapter to refer to hyperlinked pages that may be traversed from a Web browser. These files may be documents or HTML pages in the traditional sense, but may also contain "executables" in the form of applets or other techniques.

7.1 Fundamental COE Web Concepts

All Web-based segments must be DII-compliant. This applies to Web-based COE infrastructure software as well as mission-application software. The principles that govern how segments are loaded, removed, or interact with one another are the same for all DII COE segments, but COE Web component segments are treated more strictly because they are the foundation for a Web-based application.

It is important to recognize that just because a Web segment is part of the COE, it is not necessarily always present or required. Considerable flexibility is offered to customize the environment so that only the segments required to meet a specific mission application need be present at runtime. This approach allows minimization of hardware resources required to support a COE-based system.

7.1.1 COE Web Component Segments

The DII COE provides a collection of component segments to provide the architectural framework for managing and distributing data from a common Web server. Management Services include system administration, security administration, and segment registration. System administration includes the ability to monitor system performance. Security administration includes a tool for managing Web-based access control lists (consistent with the format required by the Web server), and the ability to create and manager Web user accounts.

These services are independent of any particular segment. It is anticipated that diverse segments will be able to coexist, providing access to a wide variety of data sets. However, integration and/or cooperation between segments is the responsibility of the segment developers.

7.1.1.1 Web Servers

A Web server is required to provide the interface between users and Web-based applications. The DII COE provides a Web server as a COE-component segment, thereby eliminating the requirement for individual Web segments to include a Web server. A Web mission-application segment shall *not* include its own Web server. It is required to use the Web-server segment provided by the DII COE. This is in keeping with the overall DII COE philosophy of not duplicating DII COE services.

A site installation may contain multiple platforms set aside to function as Web servers. The platforms may also serve other functions, but it is expected that sites will use firewalls to isolate Web servers from the rest of the world. For this reason, the COE requires that all Web-application segments be loaded on a machine that already contains a Web server.

7.1.1.2 Web Browsers

The COE includes a Web browser, and COE-based systems will use that browser. However, non-COE based systems can use their native browser to access services provided by the Web server. Web technology is evolving at a rapid pace, so the Web server must accommodate and address evolving Web standards. The DII COE Web server does not restrict or constrain the types of HTML files (VRML, executable content, etc.), subject to appropriate security considerations.

7.1.2 Web Mission-Application Segments

Web-application segments shall place their HTML files in the directory

```
$DATA_DIR/local/SegDir/pub
```

where *SegDir* is the segment's assigned directory. The HTML files are thus placed in the local data directory on the machine that hosts the Web Server(s). The COE creates a symbolic link from

```
COE/Comp/WebSvr/data/pub/SegDir
```

to this directory at installation time. The reason this symbolic link is created is so that the Web server can access HTML files provided by the segment. Only Web component segments are allowed to modify HTML files created by other applications, which is typically for the purpose of inserting value-added HTML tags prior to delivery to a browser. The importance of these principles cannot be overemphasized to avoid environmental conflicts between software components.

7.2 Web Account Groups

Operating systems such as Unix and NT assign individual login accounts for users. There may also be configuration files for login accounts that establish a runtime environment context. The Web environment presents a different set of requirements for user accounts since there is no need for a standard Unix or NT login account or any of the associated configuration and environmental files. Instead, Web user logins are validated by the Web server that is also responsible for enforcing access control, including restrictions based on the combination of user account and IP (or IP class) on a directory-by-directory basis.

A template is available for defining which segments and which data directories associated to a segment are available to an account group. The template shows the directory hierarchy, beginning with the directory contained in the `$DataPath` keyword in the `WebConf` segment descriptor (see Chapter 5) and cascading down through all subdirectories so that access control lists can be properly configured. It is the responsibility of the segment developer to provide documentation describing the contents of these directories to facilitate security management.

Web account groups can be used to share access privileges among a collection of users according to how they will use the system. This technique is used in the COE to identify three distinct account groups:

- Web System Administrator Accounts,
- Web Security Administrator Accounts,
- Normal Web User Accounts.

Other account groups may exist for specialized system requirements, but all account groups follow the same rules. Within a Web account group, profiles can be created as with normal COE account groups defined in Chapter 2.

7.2.1 Web Security Administrator Account

Security administration in the COE Web is implemented through a special Web account for managing the Web user account database. Precise functionality of security management is dependent on the Web server and its configuration. The role of the Web security administrator includes:

- Ability to create individual Web login accounts
- Ability to create operator Web profiles
- Ability to review the Web server error and user access logs

The Web security administrator need not be the DII security administrator, but this is recommended to centralize security management.

7.2.2 Web System Administrator Account

System administration consists of a specialized collection of functions that allow a system administrator to perform maintenance, monitoring, and configuration operations. The role of the Web system administrator includes:

- Ability to create and to restore backup tapes
- Ability to monitor and configure the Web COE-component segments
- Ability to establish site-specific products and links for user access
- Ability to review the Web server error and user access logs
- Ability to tailor Web applications (consistent with the application design) to balance overall system performance

The Web system administrator need not be the DII system administrator, but this is recommended to centralize system administration.

7.2.3 Web User Accounts

Most operators will not require, nor will Web administrators grant access to, capabilities described in the previous sections. Most system users will be performing mission-specific tasks. The precise features available depend upon which mission-application segments have been loaded and the profile assigned to the operator.

The COE establishes individual operator login accounts and stores user-specific data items, including profile information describing which options and services are available to the operator. Since users do not directly access Web segments (i.e., the Web server provides the interface between the browser and segments), many of the normal DII COE requirements for additional user-specific directories and services do not apply.

7.3 Miscellaneous

The use of server-side includes is not allowed because of the additional complexity it imposes on the Web COE in the control of data. The subsections that follow provide additional requirements and information for Web segments, beginning with HTML specifications.

7.3.1 HTML Specification

The rapid pace of innovation in Web technology makes it difficult to standardize on the exact HTML syntax that Web-application segments must support. Indeed, any HTML standard is only as good as the browser implementation. HTML version 3.2 is the latest standard, but it is not fully featured. For example, it lacks the `<FRAMES>` tag. Furthermore, version 3.2 is not fully supported by all popular browsers (e.g., Netscape 3.0 does not support style sheets). DII COE Web-server segments must, as a minimum, support HTML 3.2 and frames. The application segments should be designed to work with browsers that do not support frames or all parts of the HTML 3.2 specification, or at a minimum notify “disadvantaged” users.

An HTML file consists of a document head and a document body, as identified by the HTML tags `<HEAD> </HEAD>` and `<BODY> </BODY>`. For the purposes of this section, it is convenient to separately discuss the data content within these tags.

7.3.1.1 HTML <HEAD>

The HTML head shall contain three important data elements:

- Title (determined by the Web segment that creates the HTML file)
- Key words (used by Web search engines to identify and index Web sites for global search)
- Expiration date (using `EXPIRES`) to assist browsers in automatically rejecting out-of-date information

Key words or subjects are appended to META tags and significantly facilitate the ability of Web search engines to locate data services at other Web sites. These tags must not contain classified information (even though the entire system is running on a secure network); access to the underlying data will only be granted to users with valid accounts at the associated Web site. The use of Web search technology (bots, crawlers, spiders, etc.) requires coordination with each Web site since a login/password is required for any DII-compliant Web server connection; importantly, access to data by search engines can be provided for HEAD-only information (once a login and password have been authenticated for the special “HEAD-only” account). Additional restrictions can be implemented using access control lists in each directory. A segment that only generates dynamic, on-the-fly, HTML files may create a static HTML file with identification information specifically for the purpose of identifying the segment's information content. This HTML file shall be placed in the directory specified by the `$DataPath` keyword in the `WebConf` segment descriptor. The HTML file shall be called ***segment_name*****.htm**. The format of this HTML file shall be a standard HTML file with META tags for key words and subjects, thereby allowing HEAD-only searches to gather profile information.

7.3.1.2 HTML <BODY>

The DII COE approach is to specify the minimum set of HTML tags that are currently supported, or likely to be supported, by the popular browsers (e.g., from Microsoft and Netscape). The COE does not explicitly prohibit the use of additional HTML tags as required by a Web segment to satisfy its requirements, but provision may be made by the segment developer to alert "disadvantaged" users to potential problems.

Each Web segment is responsible for properly classifying every HTML page that it creates. The classification marking should be placed at the top and bottom of the HTML page (there is no notion of page breaks in HTML).

7.3.2 User Interface

Innovations to the Web interface offer improved user interaction and navigation via the FRAME tag, Java, JavaScript, and ActiveX functionality. These techniques enhance the user interface capabilities of Web-based applications, but at a price. The security community has expressed concerns about the potential for viruses or other malicious software spread through Java applets and applications. Developers should note that DISA is presently formulating a policy on Java usage for creating applets, and for execution by Java Virtual Machines. An update will be issued when an appropriate policy and guidance have been formulated.

Refer to the *DII Style Guide* for further style-related guidance in developing Web-based applications.

7.4 Installing Web Mission-Application Segments

Installation of Web segments, whether they are COE-component segments or mission-application segments, is accomplished like all other segments. There are some special considerations for Web mission-application segments.

Web mission-application segments must reside on the same platform as a Web Server. The COE installation tools will not allow a Web-application segment to be loaded unless there is a Web-server segment already loaded.

During installation of a Web mission-application segment, two symbolic links for use by the Web server are established, namely

- A link for accessing Web pages from the directory
  `COE/Comp/WebSvr/data/pub/`*`SegDir`*
  to the directory
  `$DATA_DIR/local/`*`SegDir`*`/pub`

- A link for accessing CGI programs from the directory
  `COE/Comp/WebSvr/data/pub/cgi-bin/`*`SegDir`*
  to the directory
  `$DATA_DIR/local/`*`SegDir`*`/cgi-bin`

Also, the `httpd.conf` file will contain an “execution” statement and a “pass” statement of the form:

```
Exec /cgi-bin/* /h/COE/Comp/WebSvr/data/pub/cgi-bin/*
Pass /*   /h/COE/Comp/WebSvr/data/pub/*
```

Here are two examples to clarify the navigation process for locating HTML files and CGI programs. Suppose a segment called `MYSEG` uses a gateway program called `TEST`, which is referenced in an HTML page as

```
FORM ACTION=/cgi-bin/MYSEG/TEST
```

This program will be found by the Web server as follows. First, the “execution” statement is used to convert the file’s location to

```
/h/COE/Comp/WebSvr/data/pub/cgi-bin/MYSEG/TEST
```

Then, the symbolic link transfers this reference to

```
$DATA_DIR/local/MYSEG/cgi-bin/TEST
```

As a second example, suppose an HTML page contains a hyperlink to a file

```
HREF=http://hostname:9000/MYSEG/DOC
```

Once the connection is established to a DII-compliant Web server, then the "pass" statement is used to convert the location of the HTML file to

```
/h/COE/Comp/WebSvr/data/pub/MYSEG/DOC
```

Then, the symbolic link transfers this reference to

```
$DATA_DIR/local/MYSEG/pub/DOC
```

Note: The DII COE establishes the SUID for the Web server. Applications must not be created which depend upon a particular setting. Instead, segments shall allow the COE segment installer to handle such details automatically.

All HTML files in `$DATA_DIR/local/`*`SegDir`*`/pub` must be readable by the Web server. The Segment Installer will automatically set the permissions on Web HTML files when the segment is loaded. Furthermore, all HTML files created by the segment for Web access must be placed in `$DATA_DIR/local/`*`SegDir`*`/pub` and must be readable by the Web server.

7.5 Supported Configurations

The COE Web component segments establish an open architecture that is not tied to a specific Web browser. They use industry standards for interfacing to the Web server (e.g., CGI) and de facto standards for HTML (as contained in HTML 3.2 and extended by the leading browsers). The HTML specification has not progressed to the point where a common presentation is guaranteed across all popular browsers.

The list of supported Web servers and Web browsers is heavily dependent on market forces as the Web industry evolves to satisfy commercial requirements. In general, it is desirable to minimize any specific dependencies on a particular browser or server. Presently, there is no commercial agreement on Web server standardization and much work remains to evaluate the leading candidates. Refer to the DISA DII COE Chief Engineer for the current status on server and browser requirements.

Precise hardware requirements in terms of memory, disk space, etc. is a function of many factors and cannot be specified in a general context. Refer to the DISA DII COE Chief Engineer for hardware configuration options.